

C. AMENDMENTS TO THE CLAIMS

1. (Original) A computer implemented method for automatically nullifying variables in a middleware computer program, said method comprising:

reading one or more variables included in one or more activation records included in the computer program;

identifying a program statement in the program where the variable is last used; and

inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable.

2. (Original) The method of claim 1 wherein the reading, identifying, and inserting are each performed by a compiler.

3. (Original) The method of claim 1 further comprising:

writing the activation records, program statement, and nullification statement to a resulting code file.

4. (Original) The method of claim 1 wherein at least one of the variables reference an object stored in a garbage collected memory heap.

5. (Original) The method of claim 1 wherein the activation records include one or more local variable definitions.

6. (Original) The method of claim 1 wherein the activation records include one or more argument parameters.

7. (Original) The method of claim 1 wherein the objects are stored in a garbage collected heap stored in a computer memory, the method further comprising:

executing a garbage collection program;

identifying, by the garbage collection program, one of the objects that was previously referenced by one of the variables included in the nullification statement; and

reclaiming the memory occupied by the identified object.

8. (Original) The method of claim 1 further comprising:

executing a compiler to perform the reading, identifying and inserting;

writing a plurality of program statements, including the program statement, to a resulting code file;

writing the nullification statement to the resulting code file in a position subsequent to the identified program statement.

9. (Original) The method of claim 8 further comprising:

identifying one or more statements from the plurality of statements where one or more other objects are last used; and

writing nullification statements for each of the other objects following the identified statement corresponding to the object's last use to the resulting code file.

10. (Original) An information handling system comprising:

one or more processors;

a memory accessible by the processors;

a middleware software application that runs on the operating system, the middleware application including a garbage-collected heap; and

a nullification tool for nullifying program references, the nullification tool comprising steps effective to:

- read one or more variables included in one or more activation records included in the computer program;
- identify a program statement in the program where the variable is last used; and
- insert a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable..

11. (Original) The information handling system of claim 10 wherein the nullification tool is a compiler.

12. (Original) The information handling system of claim 10, wherein the nullification tool is further effective to:

write the activation records, program statement, and nullification statement to a resulting code file.

13. (Original) The information handling system of claim 10 wherein at least one of the variables reference an object stored in a garbage collected memory heap.

14. (Original) The information handling system of claim 10 further comprising a garbage collected heap stored in the memory, wherein the steps are further effective to:

execute, by the processors a garbage collection program;

identify, by the garbage collection program, one of the objects that was previously referenced by one of the variables included in the nullification statement; and

reclaim the memory occupied by the identified object.

15. (Original) A computer program product stored in a computer operable media for automatically nullifying variables in a middleware computer program, said computer program product comprising:

means for reading one or more variables included in one or more activation records included in the computer program;

means for identifying a program statement in the program where the variable is last used; and

means for inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable.

16. (Original) The computer program product of claim 15 wherein the means for reading, means for identifying, and means for inserting are each performed by a compiler.

17. (Original) The computer program product of claim 15 further comprising:

means for writing the activation records, program statement, and nullification statement to a resulting code file.

18. (Original) The computer program product of claim 15 wherein at least one of the variables reference an object stored in a garbage collected memory heap.

19. (Original) The computer program product of claim 15 wherein the activation records include one or more local variable definitions.

20. (Original) The computer program product of claim 15 wherein the activation records include one or more argument parameters.

21. (Original) The computer program product of claim 15 wherein the objects are stored in a garbage collected heap stored in a computer memory, the method further comprising:

means for executing a garbage collection program;

means for identifying, by the garbage collection program, one of the objects that was previously referenced by one of the variables included in the nullification statement; and

means for reclaiming the memory occupied by the identified object.

22. (Original) The computer program product of claim 15 further comprising:

means for executing a compiler to perform the reading, identifying and inserting;

means for writing a plurality of program statements, including the program statement, to a resulting code file;

means for writing the nullification statement to the resulting code file in a position subsequent to the identified program statement.

23. (Original) The computer program product of claim 15 further comprising:

means for identifying one or more statements from the plurality of statements where one or more other objects are last used; and

means for writing nullification statements for each of the other objects following the identified statement corresponding to the object's last use to the resulting code file.

24. (Original) A method for automatically nullifying variables in a middleware computer program, said method comprising:

reading one or more variables included in one or more activation records included in the computer program;

identifying a program statement in the program where the variable is last used;

inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable;

writing a plurality of program statements, including the identified program statement, to a resulting code file; and

writing the nullification statement to the resulting code file in a position subsequent to the identified program statement.

25. (Original) An information handling system comprising:
 - one or more processors;
 - a memory accessible by the processors;
 - a middleware software application that runs on the operating system, the middleware application including a garbage-collected heap; and
 - a nullification tool for nullifying program references, the nullification tool comprising steps effective to:
 - read one or more variables included in one or more activation records included in the computer program;
 - identify a program statement in the program where one of the variables is last used;
 - insert a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable;
 - write a plurality of program statements, including the identified program statement, to a resulting code file; and
 - write the nullification statement to the resulting code file in a position subsequent to the identified program statement.

26. (Original) A computer program product stored in a computer operable media for automatically nullifying objects in a middleware computer program, said computer program product comprising:

means for reading one or more variables included in one or more activation records included in the computer program;

means for identifying a program statement in the program where the variable is last used;

means for inserting a nullification statement after the identified program statement, the nullification statement adapted to nullify the identified last-used variable;

means for writing a plurality of program statements, including the identified program statement, to a resulting code file; and

means for writing the nullification statement to the resulting code file in a position subsequent to the identified program statement.